

Please type a plus sign (+) inside this box



1120-00

A

PTO/SB/05 (11-00)

Approved for use through 10/31/2002 OMB 0651-0032

U.S. Patent and Trademark Office U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number

**UTILITY  
PATENT APPLICATION  
TRANSMITTAL**

(Only for new nonprovisional applications under 37 CFR 1.53(b))

Attorney Docket No.

005127.P001

First Inventor

Virgil Wilkins

Title

INTEGRATED INPUT/OUTPUT CONTROLLER

Express Mail Label No.

EL466330618US

**APPLICATION ELEMENTS**

See MPEP chapter 600 concerning utility patent application contents

ADDRESS TO:

Assistant Commissioner for Patents  
Box Patent Application  
Washington, DC 20231

1. ☒ Fee Transmittal Form (e.g., PTO/SB/17)  
(Submit an original and a duplicate for fee processing)

2. ☒ Applicant claims small entity status.  
See 37 CFR 1.27.

3. ☒ Specification [Total Pages 46]  
(preferred arrangement set forth below)
- Descriptive title of the Invention
  - Cross References to Related Applications
  - Statement Regarding Fed sponsored R & D
  - Reference to sequence listing, a table, or a computer program listing appendix
  - Background of the Invention
  - Brief Summary of the Invention
  - Brief Description of the Drawings (if filed)
  - Detailed Description
  - Claim(s)
  - Abstract of the Disclosure

4. ☒ Drawing(s) (35 U.S.C. 113) [Total Sheets 7]

5. Oath or Declaration [Total Pages 5]
- a. ☐ Newly executed (original or copy)
  - b. ☐ Copy from a prior application (37 C.F.R. § 1.63(d))  
(for continuation/divisional with Box 18 completed)
  - i. ☐ **DELETION OF INVENTOR(S)**  
Signed statement attached deleting inventor(s)  
named in the prior application, see 37 CFR  
1.63(d)(2) and 1.33(b)

6. ☐ Application Data Sheet. See 37 CFR 1.76

7. ☐ CD-ROM or CD-R in duplicate, large table or  
Computer Program (Appendix)

8. Nucleotide and/or Amino Acid Sequence Submission  
(if applicable, all necessary)

- a. ☐ Computer Readable Form (CRF)
- b. Specification Sequence Listing on:
  - i. ☐ CD-ROM or CD-R (2 copies); or
  - ii. ☐ paper
- c. ☐ Statements verifying identity of above copies

**ACCOMPANYING APPLICATION PARTS**

9. ☐ Assignment Papers (cover sheet & document(s))
10. ☐ 37 C.F.R. § 3.73(b) Statement ☐ Power of Attorney  
(when there is an assignee)
11. ☐ English Translation Document (if applicable)
12. ☐ Information Disclosure Statement (IDS)/PTO-1449 ☐ Copies of IDS  
Citations
13. ☐ Preliminary Amendment
14. ☒ Return Receipt Postcard (MPEP 503)  
(Should be specifically itemized)
15. ☐ Certified Copy of Priority Document(s)  
(if foreign priority is claimed)
16. ☐ Request and Certification under 35 U.S.C. 122 (b)(2)(B)(i).  
Applicant must attach form PTO/SB/35 or its equivalent.
17. ☐ Other: \_\_\_\_\_

18. If a CONTINUING APPLICATION, check appropriate box, and supply the requisite information below and in a preliminary amendment:

☐ Continuation ☐ Divisional ☐ Continuation-in-part (CIP) of prior application No: \_\_\_\_\_

Prior application Information: Examiner \_\_\_\_\_

Group/Art Unit: \_\_\_\_\_

For CONTINUATION OR DIVISIONAL APPS only: The entire disclosure of the prior application, from which an oath or declaration is supplied under Box 5b, is considered a part of the disclosure of the accompanying continuation or divisional application and is hereby incorporated by reference. The incorporation can only be relied upon when a portion has been inadvertently omitted from the submitted application parts.

**18. CORRESPONDENCE ADDRESS**

☒ Customer Number of Bar Code Label



08791

PATENT TRADEMARK OFFICE

(Insert Customer No. or Attach bar code label here)

or ☐ Correspondence address below

Name

Address

City

State

Zip Code

Country

Telephone

Fax

Name (Print/Type)

William E. Atford

Registration No. (Attorney/Agent)

37,764

Signature

*William E. Atford*

Date 11/17/00

Burden Hour Statement This form is estimated to take 0.2 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for Patents, Box Patent Application, Washington, DC 20231

# FEE TRANSMITTAL for FY 2000

Patent fees are subject to annual revision

TOTAL AMOUNT OF PAYMENT (\$ ) 883.00

## Complete if Known

Application Number  
Filing Date  
First Named Inventor Virgil Wilkins  
Examiner Name  
Group/Art Unit  
Attorney Docket No. 005127.P001

JOE29 U.S. PRO  
09/11/00

## METHOD OF PAYMENT (check one)

1. ☒ The Commissioner is hereby authorized to charge indicated fees and credit any overpayments to.

Deposit Account Number 02-2666

Deposit Account Name Blakely, Sokoloff, Taylor & Zafman LLP

☒ Charge Any Additional Fee(s) Required  
Under 37 CFR §§ 1.16, 1.17, 1.18 and 1.20

☒ Applicant claims small entity status  
See 37 CFR 1.27

☒ Payment Enclosed:

☒ Check ☐ Credit card ☐ Money Order ☐ Other

## FEE CALCULATION

### 1. BASIC FILING FEE

Large Entity	Small Entity	Fee Code	Fee (\$)	Fee Code	Fee (\$)	Fee Description	Fee Paid
101	710	201	355			Utility filing fee	355.00
106	320	206	160			Design filing fee	
107	490	207	245			Plant filing fee	
108	710	208	355			Reissue filing fee	
114	150	214	75			Provisional filing fee	
SUBTOTAL (1)							(\$ ) 355.00

### 2. EXTRA CLAIM FEES

Large Entity	Small Entity	Fee Code	Fee (\$)	Fee Code	Fee (\$)	Fee Description	Fee Paid
103	18	203	9			Claims in excess of 20	
102	80	202	40			Independent claims in excess of 3	
104	130	204	135			Multiple Dependent claim, if not paid	
109	80	209	40			**Reissue independent claims over original patent	
110	18	210	9			**Reissue claims in excess of 20 and over original patent	
SUBTOTAL (2)							(\$ ) 528.00

\*\*or number previously paid, if greater, For Reissues, see below

Large Entity	Small Entity	Fee Code	Fee (\$)	Fee Code	Fee (\$)	Fee Description	Fee Paid
103	18	203	9			Claims in excess of 20	
102	80	202	40			Independent claims in excess of 3	
104	130	204	135			Multiple Dependent claim, if not paid	
109	80	209	40			**Reissue independent claims over original patent	
110	18	210	9			**Reissue claims in excess of 20 and over original patent	
SUBTOTAL (2)							(\$ ) 528.00

## FEE CALCULATION (continued)

### 3. ADDITIONAL FEE

Large Entity	Small Entity	Fee Code	Fee (\$)	Fee Code	Fee (\$)	Fee Description	Fee Paid
105	130	205	65			Surcharge - late filing fee or oath	
127	50	227	25			Surcharge - late provisional filing fee or cover sheet	
139	130	139	130			Non-English specification	
147	2,520	147	2,520			For filing a request for reexamination	
112	920*	112	920*			*Requesting publication of SIR prior to Examiner action	
113	1,840*	113	1,840*			*Requesting publication of SIR after Examiner action	
115	110	215	55			Extension for response within first month	
116	390	216	195			Extension for response within second month	
117	890	217	445			Extension for response within third month	
118	1,390	218	695			Extension for response within fourth month	
128	1,890	228	945			Extension for response within fifth month	
119	310	219	155			Notice of Appeal	
120	310	220	155			Filing a brief in support of an appeal	
121	270	221	135			Request for oral hearing	
138	1,510	138	1,510			Petition to institute a public use proceeding	
140	110	240	55			Petition to revive - unavoidable	
141	1,240	241	620			Petition to revive - unintentional	
142	1,240	242	620			Utility issue fee (or reissue)	
143	440	243	220			Design issue fee	
144	600	244	300			Plant issue fee	
122	130	122	130			Petitions to the Commissioner	
123	130	123	130			Petitions related to provisional applications	
126	180	126	180			Submission of Information Disclosure Stmt	
581	40	581	40			Recording each patent assignment per property (times number of properties)	
146	710	246	355			Filing a submission after final rejection (37 CFR § 1.129(a))	
149	710	249	355			For each additional invention to be examined (37 CFR § 1.129(b))	
179	710	279	355			Request for Continued Examination (RCE)	
169	900	169	900			Request for expedited examination of a design application	

Other fee (specify) \_\_\_\_\_

Other fee (specify) \_\_\_\_\_

\* Reduced by Basic Filing Fee Paid

SUBTOTAL (3) (\$ )

## SUBMITTED BY

Name (Print/Type)

William E. Alford

Registration No.  
(Attorney/Agent)

37,764

Telephone

(714) 557-3800

Signature

Date

11/17/00

**WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2039.**

Atty. Ref.: 005127.P001

Express Mail No.: EL466330618US

UNITED STATES PATENT APPLICATION

FOR

**INTEGRATED INPUT/OUTPUT CONTROLLER**

Inventors:

**VIRGIL WILKINS**

**ROBERT HORN**

**MARC ACOSTA**

**SANJAY MATHUR**

Prepared by:

BLAKELY SOKOLOFF TAYLOR & ZAFMAN LLP  
12400 Wilshire Boulevard, Seventh Floor  
Los Angeles, CA 90025-1026  
(714) 557-3800

# INTEGRATED INPUT/OUTPUT CONTROLLER

## BACKGROUND OF THE INVENTION

### 1. Field of the Invention

The present invention relates generally to concurrent processing and mapping of multiple input/output requests and more particularly to input/output controllers to interface with an array of electronic devices such as computers, servers, networks, storage devices and the like.

### 2. Background Information

Low-level control of a single disk drive to store data on one or more spinning magnetic disks is well known. Usually a dedicated disk drive controller performs the low level control of moving the read/write heads to the appropriate track and sector to read, write, or format the spinning magnetic disks.

High-level control of a single disk drive is typically provided by software such as a software driver that interfaces the disk drive to a particular operating system of a computer. Another type of high-level control of a single disk drive is partitioning software that partitions a single disk drive into multiple volumes of storage areas.

Another type of high-level control for multiple disk drives is redundant array of individual disks (RAID) software that is

used to provide redundancy and increase reliability of the storage of data when multiple disks are available. Different levels of RAID software are available including RAID 0, RAID 1 and RAID 5, which provide different mechanisms of redundancy and reliability depending upon the system level requirements.

As the amount of data to be stored has become greater and the number of users requiring access to data has become larger, it has become a greater challenge to manage data storage. To ease data storage management, such as data backup and disaster recovery, it has become desirable to centralize data storage to a common pool of disks. The centralized data storage can be made available to multiple users over a network. Furthermore with the advent of the internet, centralized data storage provides increased data availability to a broader and larger distribution of users.

Because the amount of data has grown and the number of users requiring access to centralized data storage has increased, the high level control of multiple disk drives by software alone has become disadvantageous. A main disadvantage to high-level control by software is that it is slow and creates a performance bottleneck in the flow of data into and out of the multiple array of disk drives.

It is desirable to overcome the disadvantages of the prior art.

## BRIEF DESCRIPTION OF THE DRAWINGS

The features and advantages of the present invention will become apparent from the following detailed description of the present invention in which:

Figure 1 is a system block diagram in which the present invention is utilized.

Figure 2 is a functional block diagram of an embodiment of the present invention.

Figure 3 is a process diagram of a Host Interface subsystem of the present invention illustrating input and output packets and tables utilized by a Host Exchange Controller of the Host Interface subsystem.

Figure 4 is a process diagram for the Host Interface subsystem of the present invention illustrating input and output packets utilized by a Command Decode controller of the Host Interface subsystem.

Figure 5 is a process diagram of a Cache Manager of the present invention illustrating the input and output packets and tables utilized by the Cache Manager.

Figure 6 is a process diagram of a Disk Mapping Controller of the present invention illustrating the input and output packets utilized by the Disk Mapping Controller.

Figure 7 is a process diagram of a Disk Exchange Controller of the present invention illustrating the input and output packets utilized by the Disk Exchange Controller.

Figure 8 is a block diagram illustrating fields of a Range Operation Request packet utilized in the present invention.

Figure 9 is a command flow diagram showing a typical sequence of operation of a block read command.

Figure 10 is a command flow diagram showing a typical sequence of operation of a block write command.

Like reference numbers and designations in the drawings indicate like elements providing similar functionality.

### BRIEF SUMMARY OF THE INVENTION

The present invention is briefly summarized by the claims that follow below.



## DETAILED DESCRIPTION

In the following detailed description of the present invention, numerous specific details are set forth in order to provide a thorough understanding of the present invention. However, it will be obvious to one skilled in the art that the present invention may be practiced without these specific details. In other instances well known methods, procedures, components, and circuits have not been described in detail so as not to unnecessarily obscure aspects of the present invention.

The integrated I/O controller of the present invention integrates host and peripheral connections (such as disk or tape), media protocol controllers (such as Fiber channel, Ethernet or Infiniband protocol management), caching and buffer management hardware, block request mapping algorithms, packet management utilities and an embedded microcontroller. The integrated I/O controller performs resource workload balancing by monitoring a set of operational parameters and routing block level requests to the most appropriate resource. For example, two disk drives may both be able to satisfy the current data request. The integrated I/O controller will dispatch the command to the drive most likely to respond to the request in the least amount of time based upon the monitored parameters. The integrated I/O controller also provides improved fault

005127.P001  
Express Mail No.: EL466330618US

tolerance through its integration and reliability as well as its layers of management. The integrated I/O controller provides high-level control of a plurality of disks using hardware circuits in order to increase control efficiency and processing speeds. This high level control of the plurality of disks is integrated into one semiconductor integrated circuit. High-level I/O commands from the server are parsed by the integrated I/O controller and then mapped and routed to the appropriate peripheral (disk, tape, optical drive) in the array of peripherals available to the server. The subsequent data is then routed to/from the Server from/to the designated peripheral by the integrated I/O controller.

In a RAID implementation the integrated I/O controller makes the array of physical disk drives transparent to the server. The array of physical disk drives are grouped together to form larger Logical disk drives and are referenced by the Server as Volumes via a Logical Unit Number (LUN). Algorithms for RAID mapping are executed by the integrated I/O controller for managing data accesses to the Volumes. The integrated I/O controller uses microcoded structures and state machines operating concurrently in much of its circuitry, as opposed to firmware and software that operates sequentially. The microcoded structures and state machines used in the present

invention achieve increased performance while maintaining operational flexibility. A large local buffer is implemented to support the buffering of requested data blocks as well as cached data blocks yet to be requested by the Server or data blocks yet to be written to the physical disks.

Referring now to Figure 1, a system block diagram of a simple storage area network 100 is illustrated. The storage area network 100 is exemplary of a system in which the integrated I/O controller of the present invention is utilized and can be a local area network (LAN), a metropolitan area network (MAN) or a wide area network (WAN). In the storage area network 100, a client array 101 of clients 101A-101N read and write data to at least one disk array 102 having a plurality of disks 103. Between the client array 101 and the disk array 102 resides an I/O controller 110 with RAID functionality to provide centralized data storage and centralized data storage management for the network 100.

In order to access the plurality of disks 103, the clients 101A-101N of the client array 101 are coupled to a network. The clients 101A-101N couple to a hub/switch 106 via cabling or connection 105. One or more servers 108 of a server array 112 couple to the hub/switch 106 as part of the network via cable or

other type of network connection 107. In one embodiment, the hub/switch 106 is an Ethernet switch and the network 100 an Ethernet network. The cabling or connection 105 is usually Ethernet. The one or more servers 108 can include the I/O controller 110. Alternatively the one or more servers 108 can couple through cabling or connection 109 to a Fibre Channel switch 111 and then the I/O controller 110 through cabling or host connection 119 utilizing either Fibre Channel, Ethernet or Infiniband as the protocol for communication. When the switch 111 is utilized to connect the one or more servers 108 to the I/O controller 110, the term Fabric is used to describe this storage area network topology. The I/O controller 110 couples to the plurality of disks 103 and associated disk array 102 by way of cabling or connection 117. The cabling or connection 117 is usually Fibre Channel Arbitrated Loop, but can also be parallel SCSI or Infiniband. The cabling or connection 117 is also referred to herein as the physical media connection interface 117. The plurality of disk 103 maybe optical disks, magnetic disks, semi-magnetic disks, tape drives or other data storage devices. The clients 101A-101N of the client array 101 in one embodiment are personal computers but can be other electronic or electrical devices coupled into a network

configuration for storing and accessing data from the plurality of disks 103.

Referring now to Figure 2, a functional diagram of a printed circuit board (PCB) 200 containing an integrated I/O controller device 210 of the present invention is illustrated. The I/O controller system 110 includes the PCB 200. As illustrated in Figure 2, the PCB 200 includes the integrated I/O controller device 210, a host connector 201, at least one or more disk connectors 202, a cache buffer memory 204 a cache state table buffer memory 203, a system micro-processor 205, and program memory (such as FLASH, ROM, and SRAM) 206 for storing the microprocessor program, configuration tables and other program structures. The integrated I/O controller device 210, a single silicon integrated circuit, couples to the components of the printed circuit board 200 as shown in Figure 2. The host connector 201 is for coupling the I/O controller device 210 to one or more servers 108 of the server array 112 via the network switch 111. The plurality of disk connectors 202 are for coupling the integrated I/O controller device 210 to at least one disk array 102 having the plurality of disks 103. The cache buffer 204 provides temporary storage for all data blocks transferred to/from the plurality of Servers 112 and the plurality of disks 103. The cache state table buffer memory 203

is utilized to maintain status of each block of data retained in the cache buffer memory 204. The microprocessor 205 and program memory 206 are utilized to assist the integrated I/O controller device 210 in its initialization and configuration of the storage or disk array 102 for access of data to the plurality of disks 103.

Figure 2 also illustrates a functional block diagram of the integrated I/O controller device 210 which includes a host interface sub-system 220, a disk interface subsystem 230, a buffer manager 270, a cache manager 260, a disk mapping controller 250 and a micro-controller subsystem 240. The micro-controller subsystem 240 is coupled together with the host interface subsystem 220 and the disk interface subsystem 230 via the packet/control bus 214 to perform initialization and error handling at these interfaces. The micro-controller subsystem 240 is also coupled together with the buffer manager 270, the cache manager 260, and the disk mapping controller 250 in order to control the flow of non-user data information from the plurality of servers 108 of the server array 112 and the plurality of disks 103 of the disk array 102.

The buffer manager 270 couples to the cache buffer memory 204. The buffer manager 270 additionally couples to the disk

interface subsystem 230 and the host interface subsystem 220 by way of data channels 212 for the transfer of data to/from the buffer memory 204 from/to a host server 108 or a disk 103 of the disk array. The buffer manager 270 uses traditional arbitration methods to arbitrate between multiple concurrent access requests from/to the host interface subsystem 220, disk interface subsystem 230 and micro controller subsystem 240 to/from the cache buffer memory 204.

The cache manager 260 couples to the cache state table buffer 203 of the cache memory and manages the coherency of data blocks in the cache buffer memory 204. The cache memory for the cache buffer memory 204 and the cache state table buffer 203 may be a dynamic random access memory or other type of memory. Additionally, the cache buffer memory 204 and cache state table buffer 203 could be integrated into the integrated I/O controller device 210 but it is preferable that it be external in order to provide system flexibility.

The host interface subsystem 220 of the integrated I/O controller device 210 includes a Host Exchange Controller (HEC) 222 and a Command Decode Controller (CDC) 224. The host exchange controller 222 controls the physical connection and protocol between the one or more servers 108 of the server array

112 and the integrated I/O controller device 210. This host connection or attachment 119 to the servers 108 can be Fibre Channel, Ethernet, PCI, PCI-X or Infiniband in alternate embodiments. The command decode controller 224 parses all commands received from the one or more servers 108 of the server array 112 and extracts the data block level read, write requests while routing all other commands to the micro controller subsystem 240. All commands over the host connection are preferably in the form of the Small Computer System Interface (SCSI) Command Descriptor Block (CDB) standard.

The disk interface subsystem 230 controls the connections between the plurality of disks 103 of the disk array 102 and the integrated I/O controller device 210 and includes one or more Disk Exchange Controllers 232A-232N. A preferred embodiment of the disk interface subsystem executes the industry standard fibre channel connection protocol (FC-AL, FC-AL2 or FC-AL3) on a physical level. To send and receive commands and data with the plurality of disks 103, the integrated I/O controller device 210 uses the SCSI over Fibre Channel protocol (such as FCP, FCP2 or SCSI-3).



The micro-controller subsystem 240 handles non-data flow commands, error and exception handling events as well as system initialization.

Referring now to Figure 3, a process diagram of the Host Exchange Controller (HEC) 222 is illustrated. The host exchange controller 222 receives command and data frames from the physical media interface (Fibre Channel, Ethernet, Infiniband). Data frames received by the host exchange controller 222 are routed to the buffer manager 270 over the data channels 212. The command frames received by the host exchange controller 222, such as command frames 301 illustrated in Figure 3, are stored as command entries into a host exchange table 303. The host exchange table 303 in one embodiment can hold thousands of command entries.

Each command entry 313 in the host exchange table 303 has a tag number 311 which is either assigned or implied. In one embodiment, the tag number 311 is the same as the entry number, (i.e. the physical position of the entry), in the host exchange table 303. The physical location of the command entry in the host exchange table 303 implies the tag number and thus an actual tag number field 311 need not be stored with each entry.

Each entry 313 in the host exchange table also includes several other fields to properly track the progress of the

command through the integrated I/O controller device 210 and subsequently transfer data blocks to/from the requesting server 108. Field 305 contains network context information about the command which includes the Servers address on the network so response data can be routed properly back to the requesting server. Field 307 contains the SCSI CDB. Field 309 contains buffer pointers to physical locations in the cache buffer 204 where the data will be stored. The Cache Manager 260 assigns the proper values and stores them in field 309 of each entry 313 when the buffer space is physically allocated. Field 311 represents the unique tag number associated with this host exchange entry 313.

After the host exchange controller 222 creates the command entry 313 for a received command packet 301, a tag packet 315 containing the tag number for the received command is formed by the host exchange controller. The tag packet 315 is then sent to the Command Decode Controller 224 of host interface subsystem 220.

Referring now to Figure 4, a process diagram for the Command Decode Controller 224 is illustrated. The command decode controller 224 receives the tag packets 315 including the tag numbers pointing to SCSI CDBs' that need further processing. When its ready to process the next command, the command decode

controller 224 reads the command entry 313 associated with a tag packet 315, extracts the SCSI CDB from it, and validates the contents of the command entry. The command decode controller 224 includes hardware circuitry and a microcoded state-machine that specializes in decoding and queuing dataflow requests, such as block read and write commands. All commands which are received by the command decode controller 224 and determined to be valid, are placed into a pool or table of command queues 403. A separate command queue 403A-403N is maintained in the pool or table of command queues 403 for each volume of the disk array 112 that is accessible to the server array 112. The command decode controller 224 implements the industry defined SCSI Architectural Model (SAM) for command processing, including command ordering and queuing. If implemented as a RAID controller, the command decode controller 224 translates the Logical Block Address (LBA) of the requested volume into a Global (or generic) logical block address (GLBA) understood by other circuitry in the integrated I/O controller device 210. This GLBA along with a sector count from the SCSI CDB, the direction of the requested transfer (read or write), and tag number 311 are combined together to form a Range Operation Request (Ror) 801 which is further described below and illustrated in Figure 8. The Ror 801 is combined with a flag

field 405 to form an output packet 401 from the command decode controller 224 which is sent to the Cache Manager 260. In one embodiment of the invention the flag field 405 includes information about the source of the packet as well as routing information defining the destination or recipient of the packet. The flag field 405 can be expanded to include additional information about the process state of the command entry 313 to assist the micro-controller subsystem 240 in the event of an error condition or in the development and debug of the integrated I/O controller device 210. Any task management function required by an incoming host command, such as a request for the capabilities and configuration of the integrated I/O controller via SCSI Mode pages, is routed by the command decode controller 224 to the micro-controller subsystem 240.

Referring now to Figure 8, the format of the Range Operation request (Ror) packet 801 is illustrated. The Ror packet 801 is used to communicate command requests between functional blocks of the integrated I/O controller device 210. The Ror packet 801 includes a field 803, a field 805, a field 807 and a field 809. Field 803 contains the starting GLBA for the range of blocks to be operated upon. Field 805 of the Ror 801 contains the number of blocks to be operated upon (range) which depends upon if the Ror is generated by the cache manager

260, the command decode controller 224 or another functional blocks of the integrated I/O controller device 210. If the command is generated by the command decode controller 224, the value of the range (i.e. number of blocks to be operated upon) is generated from the sector count of the SCSI CDB. If the command is generated by the Cache Manager 260, the value of the range (i.e. number of blocks to be operated upon) represents the number of blocks in the cache buffer 204 to which access is desired in order to read or write data blocks to/from the disk array 102 and the plurality of drives 103. Field 807 indicates the type of range operation being requested, either a Read data operation or a Write data operation. Field 809 contains the tag number 311 from which the given Ror 801 is derived or associated.

Referring now to Figure 5, a process diagram for the Cache Manager 260 is illustrated. The cache manager 260 manages the state of all blocks within the cache buffer memory 204. The cache manager receives an input packet 401 including an Ror packet 801 and flags 405. The Ror 801 is extracted from the input packet 401 by the cache manager 260 and the range field 805 is read. The cache state buffer 203 includes a cache state table 501 which has entries each being associated with the data blocks stored into the cache buffer 204. The cache manager 260

then searches the entries in the cache state table 501 to determine if there are blocks of data in the cache buffer 204 which overlap the range of blocks specified by the range field 805 in the Ror 801. In the case of a read request, if the range of blocks requested by the command exists in the cache buffer 204 (i.e., a cache hit), the cache manager 260 updates the buffer pointer field 309 of the respective command entry 313 associated with this Ror 801 and then generates a response packet 505 which is sent to the host exchange controller 222. The response packet 505 includes a data valid field 507 indicating that data is valid and ready for transfer from the cache buffer 204 and the tag number 311. If the requested range of blocks are not detected in the cache buffer 204 for either a read or write request (i.e., a cache miss), the cache manager allocates buffer space in the cache buffer 204, updates the buffer pointers 309 in the host exchange entry 313 accordingly, and generates an output packet 503 including a new Ror 801 indicating the range of blocks required to satisfy the original range request of packet 401. The output packet 503 is passed to the disk mapping controller 250 for further processing to gain access to the disk array 112 because the desired data blocks were not in the cache buffer 204. The output packet 503 also contains a flag field 509 similar in function and capability to

the earlier described flag field 405 of the packet 401 from the command decode controller 224. Using caching algorithms (such as Least Recently Used (LRU), Pseudo Least Recently Used (PLRU), or Most Recently Used (MRU)) along with a data aging (such as time stamps on the blocks of data) technique, the cache manager 260 can determine which blocks within the cache buffer 204 should be retained and which blocks should be removed or overwritten first. The cache manager is not limited to these specified caching algorithms nor is it required that the cache manager perform a caching algorithm to determine how to retain or overwrite data in embodiments of the invention.

Referring now to Figure 6, a process diagram for the Disk Mapping Controller (DMC) 250 is illustrated. Generally, the disk mapping controller 250 converts requests for global logical block addresses (GLBAs) and maps them into physical block addresses. It is possible that a global logical block address can be mapped into more than one physical data block address when a redundant copy exists or data mirroring is utilized in a RAID controller application. If more than one source for the physical data block requested exists, the disk mapping controller 250 determines the best source of the data by evaluating a set of parameters and gathered statistics. These parameters and statistics are part of a cost function table 607

and can be used by the disk mapping controller to perform load balancing over the plurality of disk drives 103 of the disk array 102. In other embodiments of the present invention, the load balancing function need not be implemented. In a RAID controller application of the integrated I/O controller device 210, the disk mapping controller 250 implements in hardware and microcoded state-machines algorithms to provide RAID mapping of data blocks across the plurality of drives 103 of the disk array 102. RAID 0, RAID 1, RAID 0+1, RAID 5 and RAID 50 are all implemented and supported by the microcoded state-machine within the disk mapping controller 250. After calculating the required physical drive or drives to satisfy the Ror 801 of packet 503, the disk mapping controller 250 generates an output packet 601. The output packet 601 includes a Basic Drive Command 603 and the tag number 311 of the original host command requesting the data and is also referred to herein as a basic command packet 601. The disk mapping controller 250 generates output packets 601 which are sent to each of the Disk Exchange Controllers 232A-232N corresponding to each physical drive requiring access to satisfy the Ror of the input packet 503.

Alternatively, a linked list or a series of drive commands 615 can be formed, chained together and stored in a memory for later use when needed. The Cache Buffer memory 204 can be used



to store this linked list of drive commands 615 as illustrated in Figure 6. In case of the linked list of drive command 615, the disk mapping controller 250 sends a chained command packet 609 which includes a chained command pointer 611 and the tag number 311 to the disk exchange controllers 232A-232N. The chained command pointer 611 points to the list of drive commands 615 in the memory for processing by the Disk Exchange Controllers 232A-232N.

Figure 7 is a process diagram for each of the Disk Exchange Controllers (DEC) 232A-232N. Each disk exchange controller 232A-232N receives either a basic command packet 601 or alternatively a chained command packet 609 if chained commands are used. The basic command packet 601 includes the tag number 311 and a basic drive command 603. The chained command packet 609 includes the tag number 311 and the chained command pointer 611. If a disk exchange controller 232A-232N receives the basic command packet 601, it converts the basic drive commands 603 into SCSI command descriptor blocks (CDB) and then encapsulates them into an appropriate frame or output packet format 701 for transmission over the physical media connection interface 117 to the disks 103 of the disk array 102. If a disk exchange controller 232A-232N receives the chained command packet 609, it reads the chained command pointer accesses a stored command in a chain and

converts it from a basic drive command into a SCSI command descriptor blocks (CDB) which is then encapsulated into an appropriate frame or output packet format 701.

In one embodiment, each disk exchange controller 232A-232N implements the defined protocol for a SCSI initiator as documented in the SCSI Architectural Model specification (SAM) which is further refined by the SCSI over Fibre Channel (FCP) and Private Loop Direct Attach (PLDA) standards specifications for Fibre Channel implementations in generating the peripheral command packet or frame 701. Similarly in embodiments of the invention supporting Ethernet (SCSI over Internet Protocol) and Infiniband, each disk exchange controller 232A-232N is microcoded to satisfy the requirements of these protocols and generate the appropriate peripheral command packet or frame 701. When data is transmitted from the cache buffer 204 to the plurality of disk drives 103 of the disk array 102 as opposed to commands, the disk exchange controller 232A-232N encapsulates the data with the appropriate frame or packet format for the physical media connection interface 117 utilized. When data is received from the plurality of disk drives 102 the disk exchange controller removes any data encapsulation information required by the physical media interface connection 117 and routes the data to the cache buffer 204. The buffer pointers 309, in the

entry 313 specified by the tag number 311 associated with this operation, are utilized to specify the location in the cache buffer 204 to write the data.

Referring now to Figure 9, a flow diagram showing the sequence of operations for processing a host read command is illustrated. A host command packet 301, including the host read command, is received by the host exchange controller 222 and stored in the host exchange table 303. The host exchange controller 222 then passes a tag number 311 in a tag packet 315 to the command decode controller 224 which indicates where the new SCSI CDB 307 is located in the host exchange table 303. The Command Decode Controller 224 extracts and processes the SCSI CDB 307 as previously explained in the description of Figure 4. The command decode controller 224 then passes a packet 401 to the Cache Manager 260. The cache manager 260 operates on the packet 401 as previously explained in the description of Figure 5. Any requested data not currently located in the cache buffer 204 is requested from the Disk Mapping controller 250 by the generation of a new packet 503 including a new Ror 801. The disk mapping controller 250 operates on the packet 503 as previously described in the description of Figure 6 and passes the requests for data to one of the Disk Exchange Controllers 232A-232N by sending basic command packets 601 or the alternative chained

command packet 609. The disk exchange controller 232A-232N receiving the command packet acts as a SCSI initiator and manages the sending of one or more disk commands to the plurality of disk drives 103 and then subsequently manages the transfer of data from the disks to the cache buffer 204. When all data has been received from the plurality of disk drives 103 of the disk array 102, the Cache Manager 260 updates the state of the blocks in the Cache State Table 501. The cache manager 260 then notifies the host exchange controller 222 that data is valid in the cache by sending the valid packet 505. Upon receipt of the valid packet 505, the host exchange controller 222 initiates the transfer of data from the location specified in the cache buffer 204 by the buffer pointers 309 listed in the entry 313 in the host exchange table 303 for this command, to the server specified in the network context information field 305 in the same entry 313.

Referring now to Figure 10, a flow diagram showing the sequence of operations for processing a host write command is illustrated. A host command packet 301, including the host write command, is received by the Host exchange controller 222 and stored in the host exchange table 303. The host exchange controller 222 then passes a tag number 311 in a tag packet 315 to the command decode controller 224 which indicates where the

new SCSI CDB 307 is located. The Command Decode Controller 224 extracts and processes the SCSI CDB 307 as previously explained in the description of Figure 4. The command decode controller 224 then passes a packet 401 including Ror 801 to the Cache Manager 260. The cache manager 260 examines the Ror 801 of the packet 401 to determine how much buffer or storage space to allocate in the cache buffer 204 for processing the data. Once enough buffer space has been allocated to process the Ror 801, the cache manager 260 updates the buffer pointers 309 in the respective entry 313 in the host exchange table 303 as referenced by the tag number 311. The cache manager 260 then sends a valid packet 505 indicating that sufficient buffer space is available to receive data from the host. If write caching is disabled, these blocks of data will be immediately processed for transfer to the plurality of disk drives 103 of the disk array 102. The write operation proceeds by generating a new packet 503 to the Disk Mapping Controller 250 indicating the range of data to be written from the cache buffer 204. The disk mapping controller 250 operates on the packet 503 as previously explained in the description of Figure 6 and passes the requests to write data to one of the Disk Exchange Controllers 232A-232N by sending basic command packets 601 or the alternative chained command packet 609. The disk exchange controller 232A-232N

receiving the command packets acts as a SCSI initiator and manages the sending of one or more disk commands to the plurality of disk drives 103 and then subsequently the transfer of data from the cache buffer 204 to the disk drives 103 of the disk array 102. When all data has been transferred to the plurality of disk drives 103 of the disk array 102, the Cache Manager 260 updates the state of the blocks in the Cache State Table 501.

The present invention has many advantages over the prior art. One advantage of the present invention is that data communication rates are increased by integrating the I/O controller in hardware. Another advantage of the present invention is that data throughput is also increased by providing RAID mapping control in hardware. Still another advantage of the present invention is that the RAID mapping is flexible through programmability

While certain exemplary embodiments have been described and shown in the accompanying drawings, it is to be understood that such embodiments are merely illustrative of and not restrictive on the broad invention, and that this invention not be limited to the specific constructions and arrangements shown and described, since various other modifications may occur to those ordinarily skilled in the art. For example, the integrated I/O

controller has been described as interfacing to a disk array of a plurality of disks but can also interface to other storage devices or peripheral devices to provide I/O control.

Additionally, it is possible to implement the present invention or some of its features in hardware, firmware, middleware, software or a combination thereof where one or more instructions are provided in a processor readable storage medium such as a magnetic, optical, or semiconductor storage medium.

## CLAIMS

What is claimed is:

1        1.    An input/output controller integrated circuit  
2 comprising:  
3            a host interface subsystem to couple to a host to  
4            receive host commands and to transceive data blocks  
5            with the host in response to the host commands;  
6            and  
7            a mapping controller coupled to the host interface  
8            subsystem, the mapping controller to map logical block  
9            addresses of the host commands into physical block  
10           addresses of one or more peripherals to transceive the  
11           data blocks with the one or more peripherals.

1        2.    The integrated input/output controller of claim 1  
2 further comprising:  
3            a peripheral interface subsystem to couple to the one or  
4            more peripherals to issue peripheral commands and to  
5            transceive data blocks with the one or more peripherals in  
6            response to the one or more peripheral commands.

1        3.    An integrated input/output (I/O) controller integrated  
2 into an integrated circuit to read and write data between a host  
3 and one or more peripherals, the integrated I/O controller  
4 including circuits to perform:



5           receiving a high level I/O command from a host;  
6           parsing the high level I/O command to determine whether  
7           to read or write data;  
8           mapping the high level I/O request into one or more  
9           peripheral I/O commands, the one or more peripheral I/O  
10          commands indicating which of the one or more peripherals and  
11          which respective data locations are to be accessed; and  
12          servicing the high level I/O request by reading or  
13          writing data between the host and the one or more peripherals  
14          using the respective data locations.

1           4.    The integrated I/O controller of claim 3, further  
2           including circuits to perform:

3                   prior to servicing the high level I/O request, storing  
4                   data temporarily into an external cache buffer from the data  
5                   flow between the host and the one or more peripherals.

1           5.    The integrated I/O controller of claim 3, wherein,  
2                   each of the one or more peripheral I/O commands further  
3                   indicates the number of blocks of data to be serviced.

1           6.    The integrated I/O controller of claim 3, wherein,  
2                   mapping the high level I/O request into one or more  
3                   peripheral I/O commands includes circuits to perform  
4                           parsing a high level command from an I/O request  
5                           packet,

6                decoding the high level command and generating a  
7                range operation request in response thereto, and  
8                generating the one or more peripheral I/O commands  
9                in response to the range operation request.

1        7.    The integrated I/O controller of claim 3, wherein,  
2                each of the one or more peripherals are storage disks  
3                and each of the one or more peripheral I/O commands are Small  
4                Computer System Interface (SCSI) disk I/O commands.

1        8.    The integrated I/O controller of claim 3, wherein,  
2                the high level I/O command is a command of a Small  
3                Computer System Interface (SCSI) Command Descriptor Block  
4                (CDB) standard.

1        9.    The integrated I/O controller of claim 3, wherein,  
2                the circuits of the integrated I/O controller are hard  
3                wired circuits.

1        10.   The integrated I/O controller of claim 3, wherein,  
2                the circuits of the integrated I/O controller are  
3                microcoded circuits and state machines operating  
4                concurrently.

1        11.   The integrated I/O controller of claim 3, wherein,

2 the circuits of the integrated I/O controller are hard  
3 wired circuits, microcoded circuits and state machines  
4 operating concurrently.

1 12. The integrated I/O controller of claim 3, wherein,  
2 the circuits of the integrated I/O controller are  
3 programmable micro-controllers operating concurrently.

1 13. An input/output controller integrated into a single  
2 integrated circuit comprising:

3 a host interface subsystem to couple to a host to  
4 receive host commands and to transceive data blocks  
5 with the host in response to the host commands;

6 a peripheral interface subsystem to couple to a  
7 peripheral to issue peripheral commands and to  
8 transceive data blocks with the peripheral in response  
9 to the peripheral commands; and

10 a mapping controller coupled to the host interface  
11 subsystem and the peripheral interface subsystem, the  
12 mapping controller to map to blocks of data storage of  
13 the peripheral.

1 14. The integrated input/output controller of claim 9  
2 further comprising:

3 a micro-controller subsystem coupled to the host  
4 interface subsystem and the peripheral interface subsystem,

5 the micro-controller subsystem to perform initialization and  
6 to process errors and exception events.

1 15. The integrated input/output controller of claim 13  
2 further comprising:

3 a cache manager coupled to the host interface subsystem  
4 and the peripheral interface subsystem, the cache manager to  
5 manage entries in a cache buffer to temporarily store data of  
6 the data flow between the peripheral and the host.

1 16. The integrated input/output controller of claim 15  
2 further comprising:

3 a buffer manager coupled to the host interface  
4 subsystem, the peripheral interface subsystem, and the cache  
5 manager, the buffer manager to manage data storage in the  
6 cache buffer.

1 17. The integrated input/output controller of claim 13  
2 wherein,

3 the host interface subsystem includes a fibre channel  
4 host port to transceive data with the host using a fibre  
5 channel protocol.

1 18. The integrated input/output controller of claim 13  
2 wherein,

3 the host interface subsystem includes

4 a host exchange controller to control the physical  
5 connection and protocol of the host.

1 19. The integrated input/output controller of claim 13  
2 wherein,  
3 the host interface subsystem includes  
4 a command decode controller to decode host  
5 commands.

1 20. The integrated input/output controller of claim 13  
2 wherein,  
3 the peripheral interface subsystem includes  
4 a peripheral exchange controller to control the  
5 physical connection and protocol of the peripheral.

1 21. The integrated input/output controller of claim 13  
2 wherein,  
3 the peripheral interface subsystem includes a  
4 Fibrechannel disk port to transceive data with the peripheral  
5 using a Fibrechannel protocol.

1 22. A method of processing host write commands between a  
2 host and one or more peripherals, the method comprising:  
3 receiving a high level write command from a host, the  
4 high level write command requesting to write data from the  
5 host transparently to one or more peripherals;

6 mapping the high level write command into one or more  
7 low level peripheral write commands, the one or more low  
8 level peripheral write commands indicating data storage  
9 locations in the one or more peripherals into which data from  
10 the host is to be written;

11 servicing the high level write command by receiving data  
12 from the host for storage into the data storage locations in  
13 the one or more peripherals; and

14 writing the data from the host into the data storage  
15 locations in the one or more peripherals in response to the  
16 one or more low level peripheral write commands.

1 23. The method of claim 22 further comprising:

2 prior to writing the data from the host into the data  
3 storage locations, temporarily storing the data from the host  
4 into a buffer memory.

1 24. The method of claim 22 wherein,

2 the high level write command indicates the number of  
3 blocks of data to be written from the host.

1 25. A method of processing host read commands between a host  
2 and one or more peripherals, the method comprising:

3 receiving a high level read command from a host, the  
4 high level read command requesting to transparently read data  
5 from the one or more peripherals to the host;

6 mapping the high level read command into one or more low  
7 level peripheral read commands, the one or more low level  
8 peripheral read commands indicating data storage locations in  
9 the one or more peripherals from which data is to be accessed  
10 and read out to the host;

11 servicing the high level read request by reading data  
12 from the data storage locations in the one or more  
13 peripherals in response to the one or more low level  
14 peripheral read commands; and

15 providing the data to the host.

1 26. The method of claim 25 further comprising:

2 prior to providing the data to the host, temporarily  
3 storing the data from the one or more peripherals into a  
4 buffer memory.

1 27. The method of claim 25 wherein,

2 the high level read command indicates the number of  
3 blocks of data to be read into the host.

1 28. An integrated input/output controller comprising:

2 a semiconductor integrated circuit including,

3 a host interface to couple to one or more

4 servers to transceive data between the one or more

5 servers and the integrated input/output controller in

6 response to host commands,

7                   a mapping controller coupled to the host  
8                   interface subsystem, the mapping controller to map to  
9                   blocks of data storage of a peripheral, and  
10                  a micro-controller coupled to the host interface  
11                  subsystem, the micro-controller to perform  
12                  initialization and handle error and exception handling  
13                  events.

1           29. The integrated input/output controller of claim 28  
2 wherein,  
3                  the host interface subsystem includes a fibre channel  
4 host port to transceive data with the one or more servers  
5 using a fibre channel protocol.

1           30. The integrated input/output controller of claim 28  
2 wherein,  
3                  the host interface subsystem includes a command decode  
4 controller to receive and decode host command packets.

1           31. The integrated input/output controller of claim 30  
2 wherein,  
3                  the command decode controller to further validate a host  
4 command and to initiate execution of the host command by the  
5 integrated input/output controller.

1           32. The integrated input/output controller of claim 30  
2 wherein,



the command decode controller to further validate a host command and to queue the host command in a queue to be executed in an order, the command queue associated with a volume accessible by one of the one or more a servers.

33. The integrated input/output controller of claim 30 wherein,

the command decode controller to further validate a host command and to determine that the host command is invalid, the host command is passed to the micro-controller subsystem to process the invalidity.

34. The integrated input/output controller of claim 30 wherein,

the host command packets are high level input/output requests.

35. The integrated input/output controller of claim 28 further comprising:

a peripheral interface to couple to one or more peripheral devices and transceive data between the one or more peripheral devices and the one or more servers coupled to the host port.

36. The integrated input/output controller of claim 28 wherein,

3 the integrated input/output controller is an integrated  
4 RAID controller to transceive data between one or more disks  
5 of at least one disk array and the integrated input/output  
6 controller further comprises:

7 a disk interface to couple to the one or more  
8 disks of the at least one disk array to transceive the  
9 data to or from the one or more servers; and

10 the mapping controller is a RAID mapping controller  
11 to flexibly control the mapping of blocks of data  
12 storage on the one or more disks of the at least one  
13 disk array.

1 37. The integrated input/output controller of claim 36  
2 wherein,

3 the one or more disks of the at least one array of disks  
4 are magnetic storage media, optical storage media or  
5 semiconductor storage media.

1 38. The integrated input/output controller of claim 36  
2 wherein,

3 the disk interface subsystem includes one or more fibre  
4 channel disk ports to transceive data with the one or more  
5 disks of the at least one disk array using a fibre channel  
6 protocol.

1 39. The integrated input/output controller of claim 28  
2 wherein,

3 the mapping controller provides RAID mapping automation.

1 40. The integrated input/output controller of claim 28

2 wherein,

3 the mapping controller is programmable hardware to  
4 flexibly control the mapping of blocks of data storage on the  
5 one or more disks of the at least one disk array.

1 41. The integrated input/output controller of claim 28

2 wherein,

3 the mapping controller to receive a requested command  
4 input packet to generate expanded command output packets in  
5 response thereto, the requested command input packet  
6 functions as a logical address and the expanded command  
7 output packets function as physical addresses.

1 42. The integrated input/output controller of claim 28

2 further comprising:

3 a buffer manager and a cache manager, the buffer manager  
4 and the cache manager to couple to a cache buffer and a cache  
5 table buffer respectively to flexibly control the reading and  
6 writing of data to and from the mapped data storage on the  
7 one or more disks of the at least one disk array.

1 43. A storage area network for central data storage and

2 management, the storage area network comprising:

3 at least one server to couple to a network;

at least one disk array having a plurality of disks; and  
at least one fibre channel controller to couple to at  
least one server and at least one disk array, the at least  
one fibre channel controller to read and write data between  
the at least one server and the at least one disk array, the  
at least one fiber channel controller having  
a cache memory,  
a microprocessor to initialize the fiber channel  
controller upon power up and reset,  
a programmable random access memory (PRAM) to store  
initialization instructions in firmware, and  
a hardware redundant array of independent disks (RAID)  
controller to control the reading and writing of data between  
the at least one server and the at least one disk array.

44. The storage area network of claim 43 for central data  
storage and management, wherein the hardware RAID controller  
includes,

a host interface to couple to the at least one  
server to receive data from the at least one server for  
storage into the one or more disks of the at least one  
disk array and to transmit data to the at least one  
server for data accessed from the one or more disks of  
the at least one disk array;

a disk interface to couple to the one or more  
disks of the at least one disk array to transmit data  
to the one or more disks of the at least one disk array

13 for storage and to receive data from the one or more  
14 disks of the at least one disk array when accessed,  
15 a RAID mapping controller to flexibly control the  
16 mapping of blocks of storage on the one or more disks  
17 of the at least one disk array, and  
18 a micro-controller coupled to the host interface  
19 subsystem, the disk interface subsystem, and the RAID  
20 mapping subsystem, the micro-controller to handle non-  
21 data flow commands, error and exception handling events  
22 as well as system initialization.

1 44. The storage area network of claim 43 for central data  
2 storage and management, wherein,  
3 the cache memory to provide a cache data buffer and a  
4 cache table buffer for the integrated RAID controller  
5 integrated circuit.

1 46. The storage area network of claim 43 for central data  
2 storage and management, wherein,  
3 the plurality of disks of the at least one array of  
4 disks are magnetic storage media, optical storage media or  
5 semiconductor storage media.

1 47. An integrated input/output (I/O) controller comprising:  
2 a semiconductor integrated circuit including,  
3 a host command manager to manage high level host  
4 I/O requests from a plurality of hosts;

5 a mapping engine to map high level host I/O  
6 requests into low level I/O commands; and  
7 a low level command manager to manage data read  
8 and data write accesses into and out of a peripheral  
9 device in response to the low level I/O commands.

1 48. The integrated input/output (I/O) controller of claim 47  
2 further comprising:

3 a buffer manager to arbitrate access by the one or  
4 more servers and to control data reads and data writes  
5 into and out of a buffer memory.

1 49. The integrated input/output (I/O) controller of claim 47  
2 further comprising:

3 a micro-controller to handle non-data flow  
4 commands, system initialization and error handling  
5 exception conditions.

1 50. The integrated input/output (I/O) controller of claim 47  
2 wherein

3 the integrated I/O controller is a RAID controller  
4 and the peripheral device is a plurality of disks  
5 responsive to disk I/O commands.

1 51. A method of transceiving data to a plurality of disks in  
2 an integrated circuit, the method comprising:

3           receiving an I/O request from at least one server, the  
4   I/O request indicating a volume to access;  
5           mapping the I/O request into a disk I/O command, the  
6   disk I/O command indicating one or more disks of the  
7   plurality of disks to process the disk I/O command; and  
8           managing a data read or a data write access to the  
9   plurality of disks in response to the disk I/O command.

1           52. The method of claim 51 for transceiving data to a  
2   plurality of disks in an integrated circuit further comprising:  
3           arbitrating a data read or a data write access to a  
4   buffer memory to temporarily store the data prior to the data  
5   read or data write access to the plurality of disks.

## ABSTRACT OF THE DISCLOSURE

An integrated I/O controller in an integrated circuit is provided for centralized data management. The integrated circuit includes a host interface, a disk interface, and a mapping controller implemented in hardware to speed data processing and provide fault tolerance as exemplified with RAID configurations. The mapping controller provides block mapping across a plurality of peripherals or disk drives in a disk array. The integrated I/O controller can be utilized in storage area network systems and network area storage systems as well as other networking systems or devices.



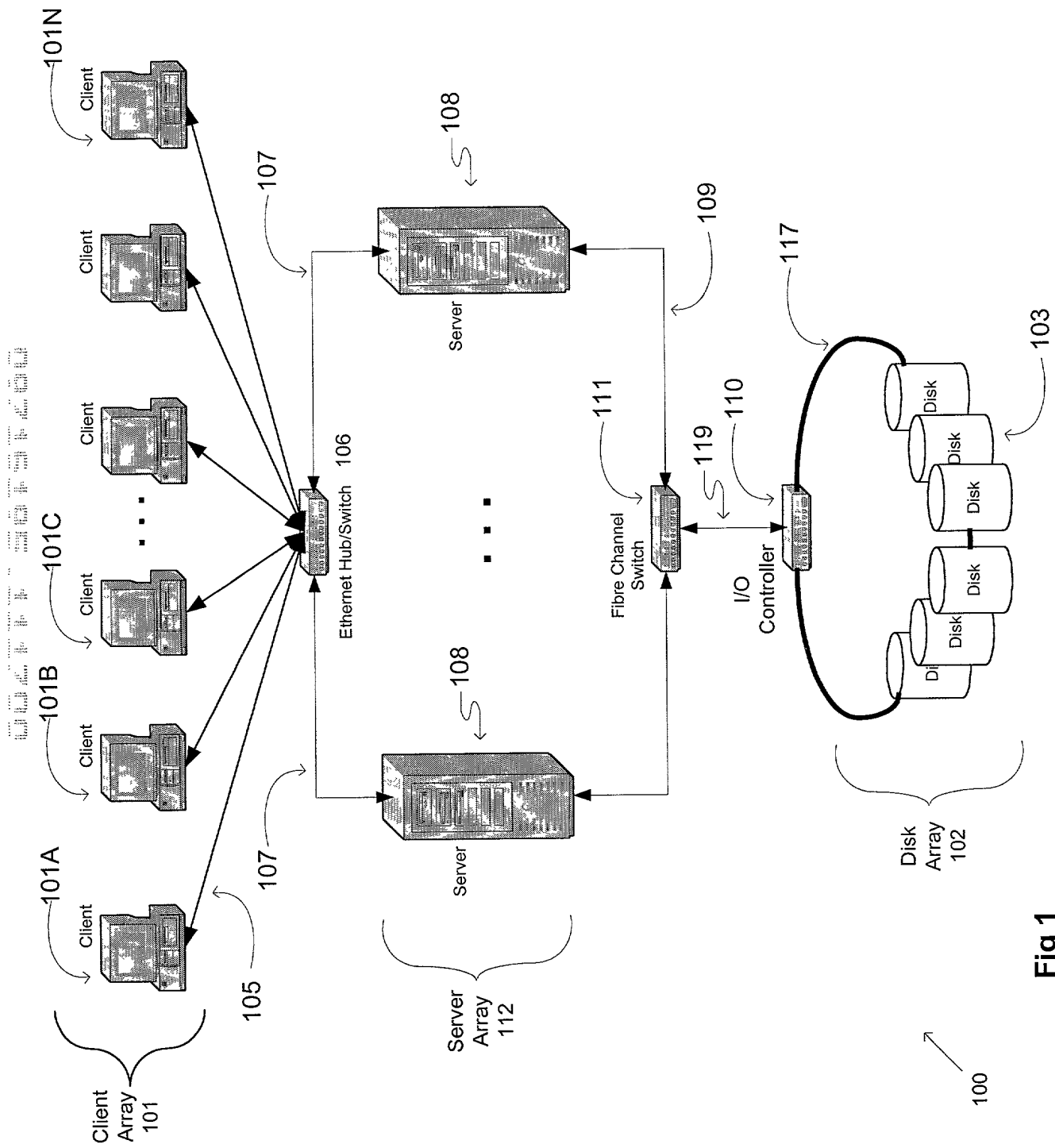
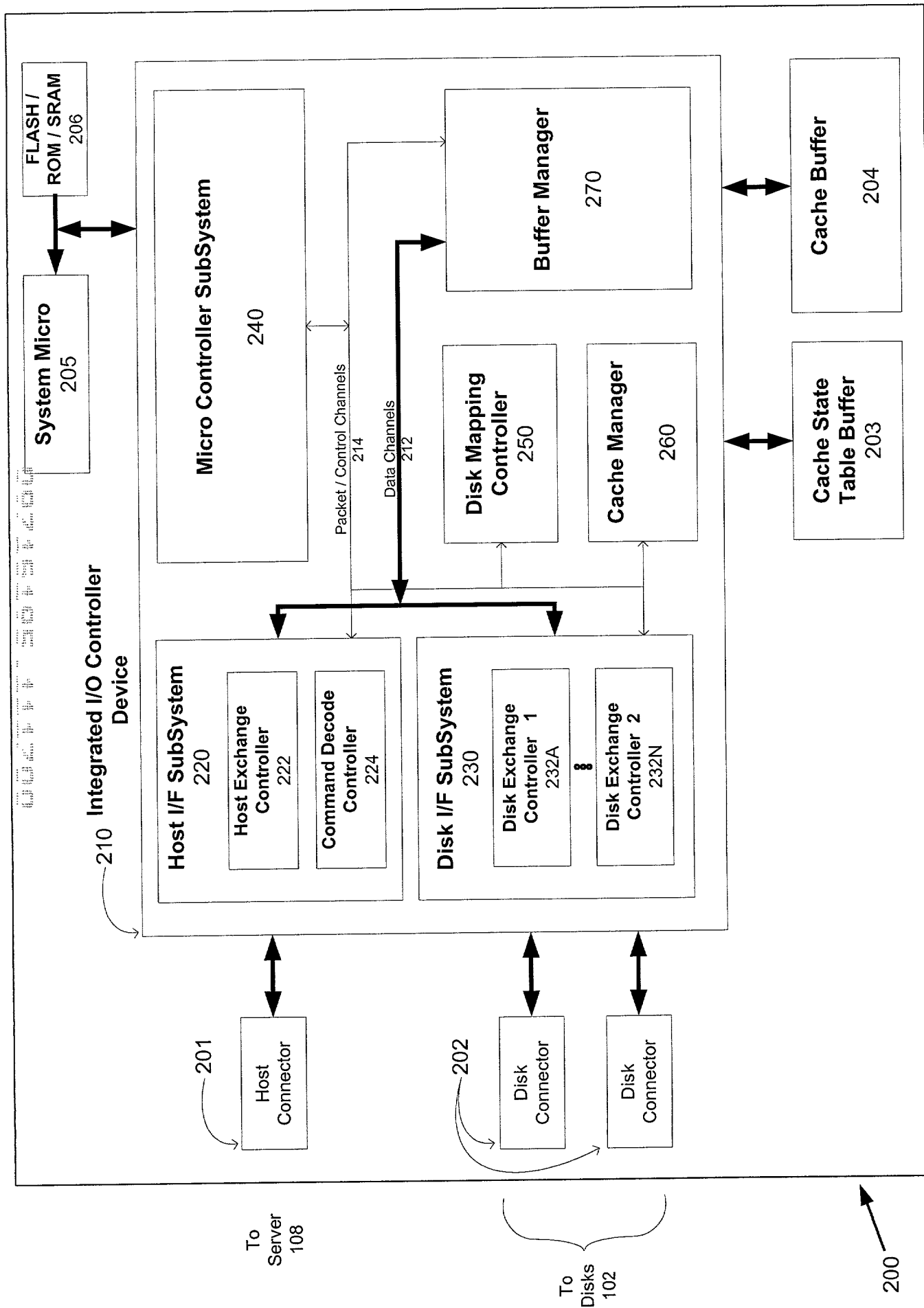


Fig 1



**Fig 2** Integrated I/O Controller PCBA Diagram

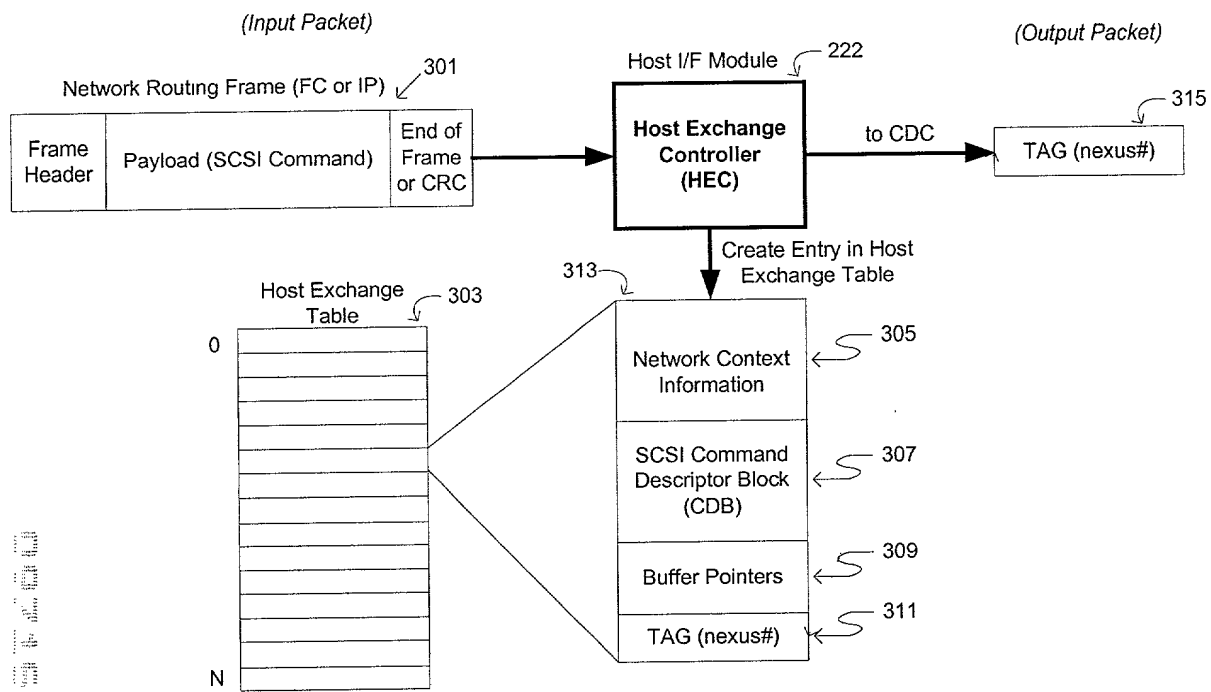


fig 3

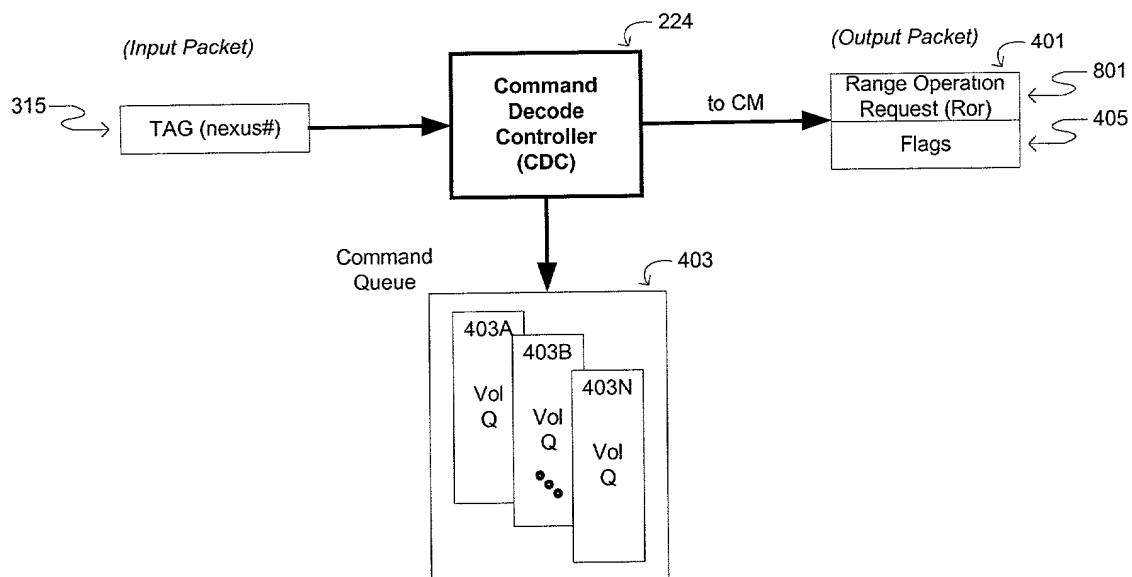


fig 4

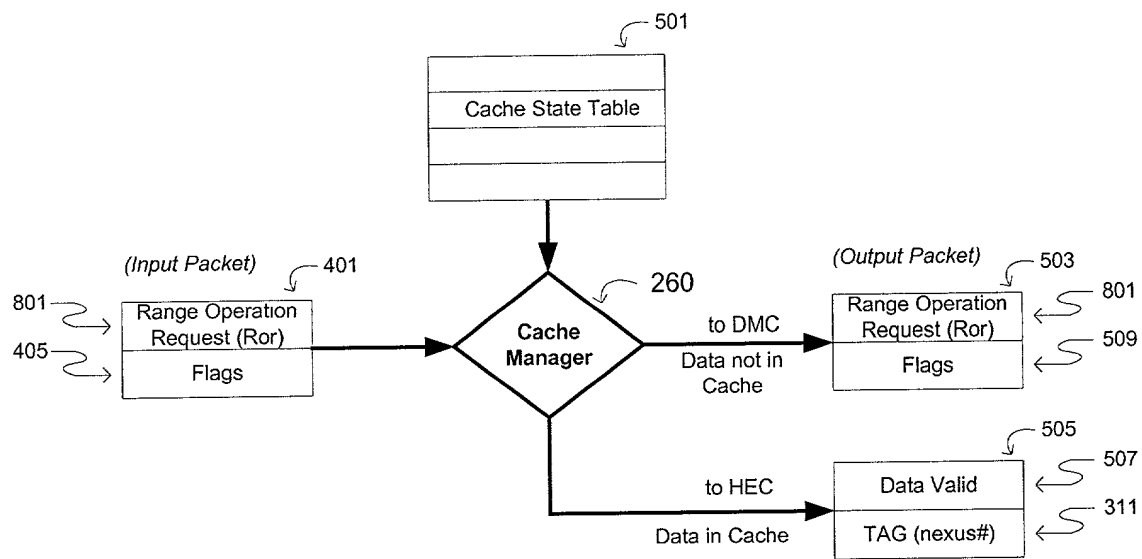


fig 5

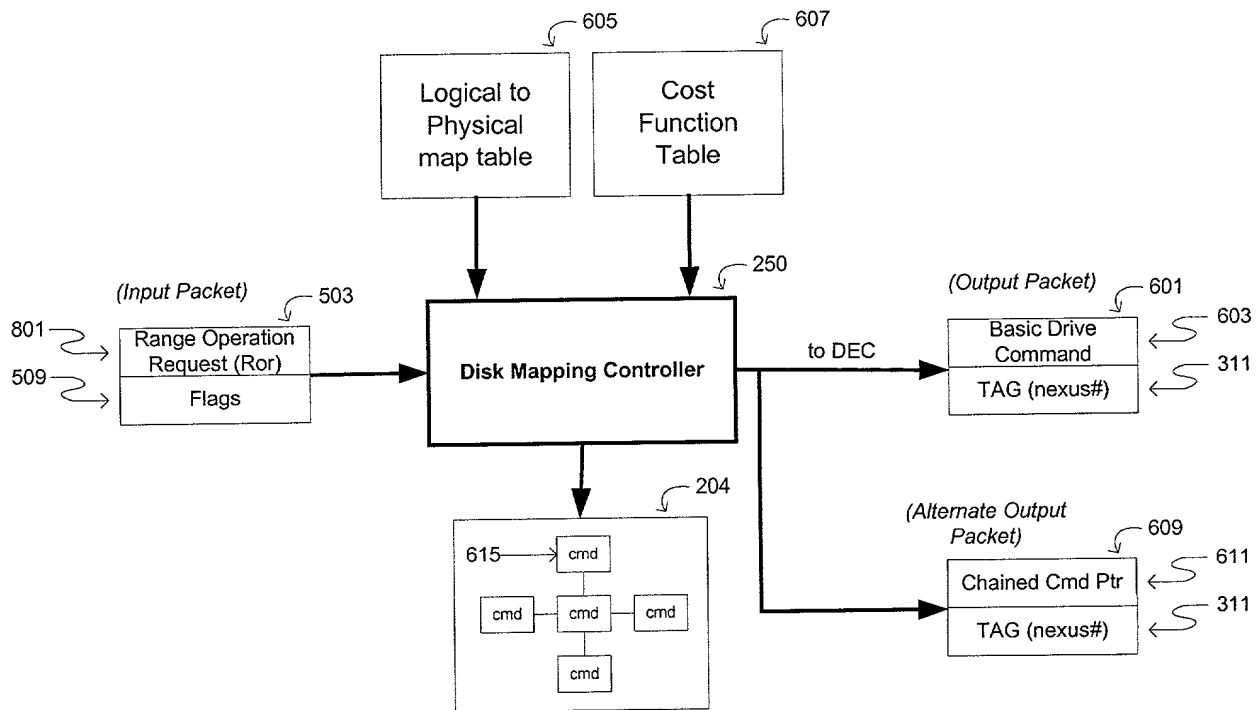


fig 6

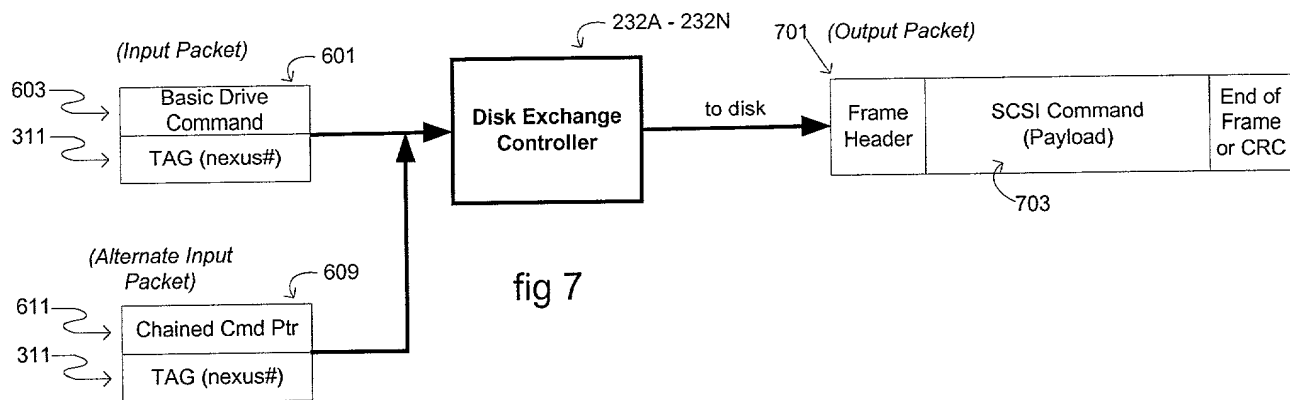


fig 7

### Range Operation Request (Ror) packet

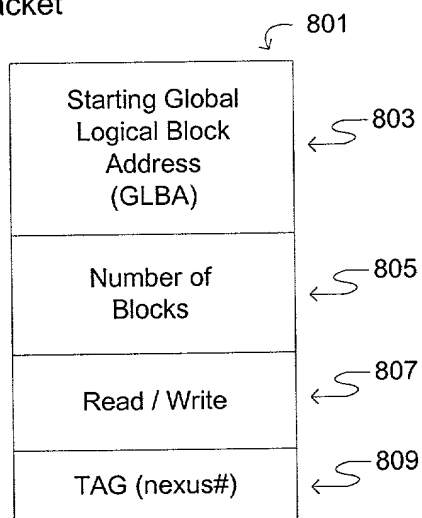


fig 8

# Host Read Command Flow

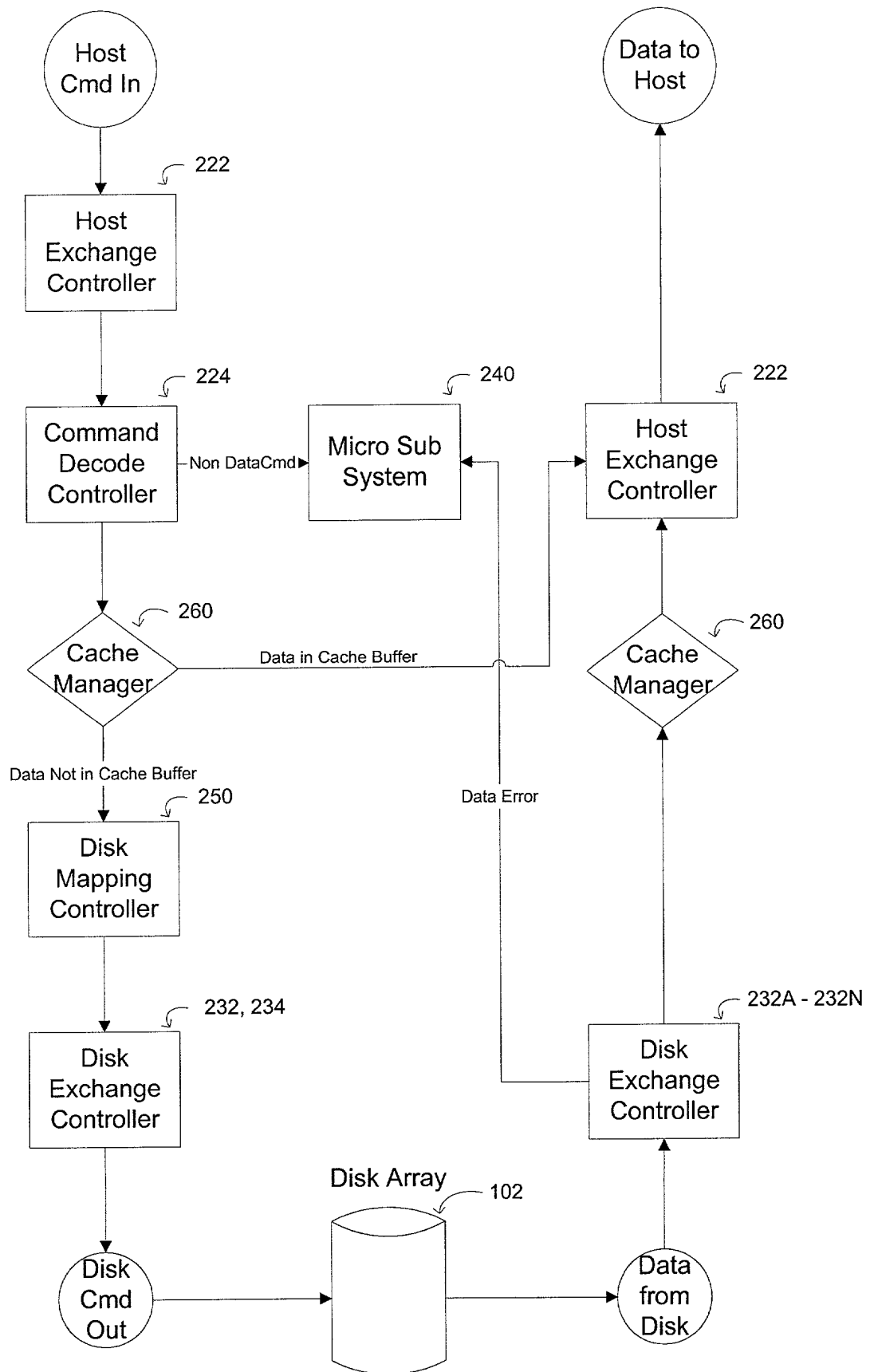


fig 9

## Host Write Command Flow

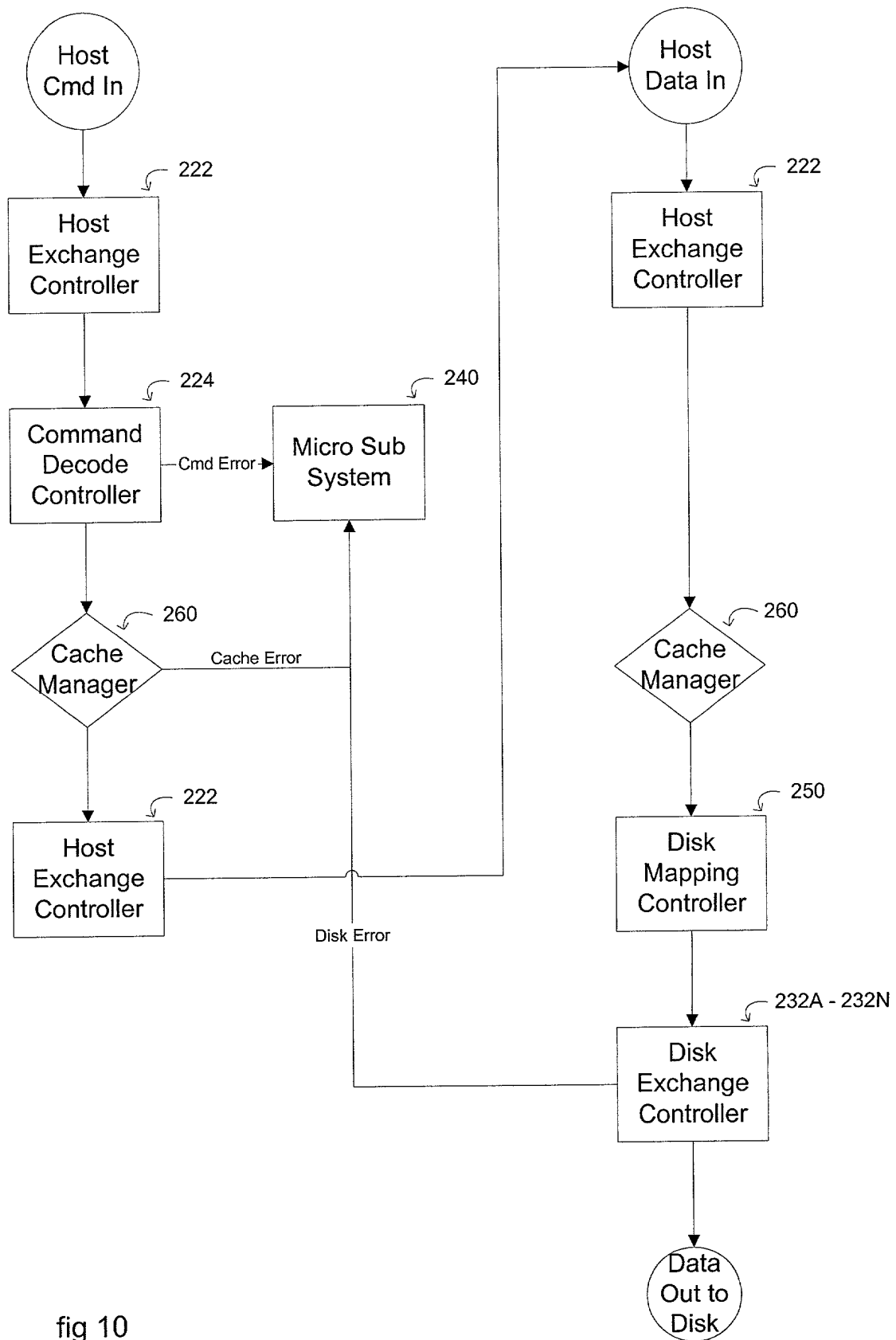


fig 10

to Disk array 102

## DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below, next to my name.

I believe I am the original, first, and sole inventor (if only one name is listed below) or any original, first, and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled

### INTEGRATED INPUT/OUTPUT CONTROLLER

the specification of which ☒ is attached hereto.

☐ was filed on \_\_\_\_\_ as \_\_\_\_\_

United States Application Number \_\_\_\_\_

or PCT International Application Number \_\_\_\_\_

and was amended on \_\_\_\_\_

(if applicable)

I hereby state that I have reviewed and understand the contents of the above-identified specification, including the claim(s), as amended by any amendment referred to above. I do not know and do not believe that the claimed invention was ever known or used in the United States of America before my invention thereof, or patented or described in any printed publication in any country before my invention thereof or more than one year prior to this application, that the same was not in public use or on sale in the United States of America more than one year prior to this application, and that the invention has not been patented or made the subject of an inventor's certificate issued before the date of this application in any country foreign to the United States of America on an application filed by me or my legal representatives or assigns more than twelve months (for a utility patent application) or six months (for a design patent application) prior to this application.

I acknowledge the duty to disclose all information known to me to be material to patentability as defined in Title 37, Code of Federal Regulations, Section 1.56.

I hereby claim foreign priority benefits under Title 35, United States Code, Section 119(a)-(d), of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

#### Prior Foreign Application(s):

APPLICATION NUMBER	COUNTRY (OR INDICATE IF PCT)	DATE OF FILING (day, month, year)	PRIORITY CLAIMED UNDER 37 USC 119
			<input type="checkbox"/> No <input type="checkbox"/> Yes
			<input type="checkbox"/> No <input type="checkbox"/> Yes
			<input type="checkbox"/> No <input type="checkbox"/> Yes

I hereby claim the benefit under Title 35, United States Code, Section 119(e) of any United States provisional application(s) listed below:

APPLICATION NUMBER	FILING DATE





**Full Name of Third/Joint Inventor** (given name, family name) Marc Acosta

Inventor's Signature \_\_\_\_\_ Date \_\_\_\_\_

Residence San Clemente, California USA Citizenship USA  
(City, State) (Country)

Mailing Address 2606 Calle Onice  
San Clemente, California 92354 USA

**Full Name of Fourth/Joint Inventor** (given name, family name) Sanjay Mathur

Inventor's Signature \_\_\_\_\_ Date \_\_\_\_\_

Residence Laguna Hills, California USA Citizenship USA  
(City, State) (Country)

Mailing Address 27212 Hidden Trail Road  
Laguna Hills, California 92653 USA

**Full Name of Fifth/Joint Inventor** (given name, family name) \_\_\_\_\_

Inventor's Signature \_\_\_\_\_ Date \_\_\_\_\_

Residence \_\_\_\_\_ Citizenship \_\_\_\_\_  
(City, State) (Country)

Mailing Address \_\_\_\_\_  
\_\_\_\_\_

**Full Name of Sixth/Joint Inventor** (given name, family name) \_\_\_\_\_

Inventor's Signature \_\_\_\_\_ Date \_\_\_\_\_

Residence \_\_\_\_\_ Citizenship \_\_\_\_\_  
(City, State) (Country)

Mailing Address \_\_\_\_\_  
\_\_\_\_\_

**Full Name of Seventh/Joint Inventor** (given name, family name) \_\_\_\_\_

Inventor's Signature \_\_\_\_\_ Date \_\_\_\_\_

Residence \_\_\_\_\_ Citizenship \_\_\_\_\_  
(City, State) (Country)

Mailing Address \_\_\_\_\_  
\_\_\_\_\_

**Full Name of Eighth/Joint Inventor** (given name, family name) \_\_\_\_\_

Inventor's Signature \_\_\_\_\_ Date \_\_\_\_\_

Residence \_\_\_\_\_ Citizenship \_\_\_\_\_  
(City, State) (Country)

Mailing Address \_\_\_\_\_  
\_\_\_\_\_

**Full Name of Ninth/Joint Inventor** (given name, family name) \_\_\_\_\_

Inventor's Signature \_\_\_\_\_ Date \_\_\_\_\_

Residence \_\_\_\_\_ Citizenship \_\_\_\_\_  
(City, State) (Country)

Mailing Address \_\_\_\_\_  
\_\_\_\_\_

**Full Name of Tenth/Joint Inventor** (given name, family name) \_\_\_\_\_

Inventor's Signature \_\_\_\_\_ Date \_\_\_\_\_

Residence \_\_\_\_\_ Citizenship \_\_\_\_\_  
(City, State) (Country)

Mailing Address \_\_\_\_\_  
\_\_\_\_\_

## Appendix A

I hereby appoint BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP, a firm including: William E. Alford, Reg. No. 37,764; Farzad E. Amini, Reg. No. 42,261; William Thomas Babbitt, Reg. No. 39,591; Carol F. Barry, Reg. No. 41,600; Jordan Michael Becker, Reg. No. 39,602; Lisa N. Benado, Reg. No. 39,995; Bradley J. Bereznak, Reg. No. 33,474; Michael A. Bernadicou, Reg. No. 35,934; Roger W. Blakely, Jr., Reg. No. 25,831; R. Alan Burnett, Reg. No. 46,149; Gregory D. Caldwell, Reg. No. 39,926; Andrew C. Chen, Reg. No. 43,544; Thomas M. Coester, Reg. No. 39,637; Donna Jo Coningsby, Reg. No. 41,684; Dennis M. deGuzman, Reg. No. 41,702; Stephen M. De Klerk, Reg. No. P46,503; Michael Anthony DeSanctis, Reg. No. 39,957; Daniel M. De Vos, Reg. No. 37,813; Sanjeet Dutta, Reg. No. P46,145; Matthew C. Fagan, Reg. No. 37,542; Tarek N. Fahmi, Reg. No. 41,402; George Fountain, Reg. No. 36,374; Paramita Ghosh, Reg. No. 42,806; James Y. Go, Reg. No. 40,621; James A. Henry, Reg. No. 41,064; Willmore F. Holbrow III, Reg. No. P41,845; Sheryl Sue Holloway, Reg. No. 37,850; George W. Hoover II, Reg. No. 32,992; Eric S. Hyman, Reg. No. 30,139; William W. Kidd, Reg. No. 31,772; Sang Hui Kim, Reg. No. 40,450; Walter T. Kim, Reg. No. 42,731; Eric T. King, Reg. No. 44,188; Erica W. Kuo, Reg. No. 42,775; George B. Leavell, Reg. No. 45,436; Gordon R. Lindeen III, Reg. No. 33,192; Jan Carol Little, Reg. No. 41,181; Kurt P. Leyendecker, Reg. No. 42,799; Joseph Lutz, Reg. No. 43,765; Michael J. Mallie, Reg. No. 36,591; Andre L. Marais, under 37 C.F.R. § 10.9(b); Paul A. Mendonsa, Reg. No. 42,879; Clive D. Menezes, Reg. No. 45,493; Chun M. Ng, Reg. No. 36,878; Thien T. Nguyen, Reg. No. 43,835; Thinh V. Nguyen, Reg. No. 42,034; Dennis A. Nicholls, Reg. No. 42,036; Daniel E. Ovanezian, Reg. No. 41,236; Kenneth B. Paley, Reg. No. 38,989; Marina Portnova, Reg. No. P45,750; William F. Ryann, Reg. No. 44,313; James H. Salter, Reg. No. 35,668; William W. Schaal, Reg. No. 39,018; James C. Scheller, Reg. No. 31,195; Jeffrey S. Schubert, Reg. No. 43,098; Jeffrey Sam Smith, Reg. No. 39,377; Maria McCormack Sobrino, Reg. No. 31,639; Stanley W. Sokoloff, Reg. No. 25,128; Judith A. Szepesi, Reg. No. 39,393; Vincent P. Tassinari, Reg. No. 42,179; Edwin H. Taylor, Reg. No. 25,129; John F. Travis, Reg. No. 43,203; Joseph A. Twarowski, Reg. No. 42,191; Thomas A. Van Zandt, Reg. No. 43,219; Lester J. Vincent, Reg. No. 31,460; Glenn E. Von Tersch, Reg. No. 41,364; John Patrick Ward, Reg. No. 40,216; Mark L. Watson, Reg. No. P46,322; Thomas C. Webster, Reg. No. P46,154; and Norman Zafman, Reg. No. 26,250; my patent attorneys, and Firasat Ali, Reg. No. 45,715; and Justin M. Dillon, Reg. No. 42,486; Raul Martinez, Reg. No. 46,904; my patent agents, with offices located at 12400 Wilshire Boulevard, 7th Floor, Los Angeles, California 90025, telephone (714) 557-3800, with full power of substitution and revocation, to prosecute this application and to transact all business in the Patent and Trademark Office connected herewith.